

1 CLAIMS

2

3 1. An application program interface (API) comprising:

4 a set of one or more residual difference data structures including residual

5 difference information for encoded multimedia content; and

6 a corresponding set of one or more control command data structures

7 including control commands to control prediction and addition of residual coding

8 information to decode multimedia content, wherein the API includes the control

9 commands necessary to control multimedia decoding in accordance with any of a

10 plurality of standard multimedia codecs.

11

12 2. An API according to claim 1, wherein the residual difference data

13 structures and the control command data structures are two of a plurality of

14 dynamically selected types of operational data structure(s) of the API.

15

16 3. An API according to claim 1, wherein the control command data

17 structures include macroblock control commands, to control prediction and

18 addition of residual coding information on a macroblock level.

19

20 4. An API according to claim 1, wherein the API selectively invokes at

21 least the subset of control commands upon identifying a processing capability of at

22 least the decoder and the accelerator.

23

24

25

1 5. An API according to claim 4, wherein the API iteratively issues a set
2 of control commands to negotiate an acceptable communication capability
3 between one or more decoder applications and one or more hardware accelerators.
4

5 6. An API according to claim 1, wherein the API negotiates an
6 acceptable communication capability between one or more decoders and one or
7 more hardware accelerators by iteratively issuing configuration commands
8 reflecting various alternative degrees and methods of decoding acceleration
9 capability until choosing one that is acceptable to both the decoder(s) and the
10 accelerator(s).
11

12 7. An API according to claim 1, further comprising:
13 data structures, generated in response to command(s) received from a
14 decoder application, consisting of deblocking filter control command(s) to control
15 one or more deblocking filter attributes of a communicatively coupled hardware
16 accelerator.
17

18 8. An API according to claim 7, wherein the deblocking filter control
19 commands control a smoothing filter across block boundaries of a decoded
20 picture.
21

22 9. An API according to claim 7, wherein the deblocking filter control
23 commands include a flag sent for each block edge denoting whether the
24 deblocking filter is to be applied across the associated block edge.
25

1 **10.** An API according to claim 1, wherein the control command data
2 structure is a fixed-size data structure for each macroblock of a picture.

3
4 **11.** An API according to claim 10, wherein the API utilizes an absolute
5 macroblock address within each control command data structure to specify which
6 macroblock to process, facilitating independent processing of each individual
7 macroblock control command of a picture.

8
9 **12.** An API according to claim 11, wherein the absolute macroblock
10 address facilitates parallel processing of two or more macroblocks of a picture.

11
12 **13.** An API according to claim 11, wherein the API utilizes a data
13 location pointer within each control command data structure to specify the
14 location within the corresponding residual difference data buffer for the data
15 associated with the macroblock control command, facilitating independent
16 processing of each individual macroblock control command of a picture.

17
18 **14.** An API according to claim 13, wherein the data location pointer
19 facilitates parallel processing of two or more macroblocks of a picture.

20
21 **15.** A storage medium comprising a plurality of executable instructions
22 which, when executed, implement an application program interface (API)
23 according to claim 1.
24
25

1 **16.** A computing system comprising:
2 a storage medium including a plurality of executable instructions; and
3 an execution unit, coupled to the storage medium, to execute at least a
4 subset of the executable instructions to implement an application program
5 interface (API) according to claim 1.

6
7 **17.** An application program interface (API) comprising:
8 one or more auto-negotiation data structures, dynamically generated by the
9 API to negotiate at least a set of processing standards among and between one or
10 more elements of a media processing system; and
11 one or more operational data structures, dynamically generated by the API
12 to support processing of media content among and between the media processing
13 system elements in accordance with the negotiated processing standard(s).

14
15 **18.** An API according to claim 17, the operational data structures
16 comprising:
17 a set of one or more residual difference data structures including residual
18 difference information for encoded multimedia content; and
19 a set of one or more control command data structures including control
20 commands to control prediction and addition of residual coding information to
21 decode multimedia content, wherein the API includes the control commands
22 necessary to control multimedia decoding in accordance with any of a plurality of
23 multimedia codecs and invokes at least a subset of the control commands to
24 interface the decoder with the multimedia accelerator.

1 **19.** An API according to claim 17, the operational data structures
2 comprising:

3 a raw bitstream data structure, dynamically generated to transfer raw media
4 content bitstream(s) between media processing system elements.
5

6 **20.** An API according to claim 17, wherein the auto-negotiation data
7 structure(s) are dynamically generated to negotiate a split in media processing
8 between identified media processing system elements.
9

10 **21.** An API according to claim 20, the operational data structures
11 comprising:

12 a set of one or more residual difference data structures including residual
13 difference information for encoded multimedia content; and

14 a set of one or more control command data structures including control
15 commands to control prediction and addition of residual coding information to
16 decode multimedia content, wherein the API includes the control commands
17 necessary to control multimedia decoding in accordance with any of a plurality of
18 multimedia codecs and invokes at least a subset of the control commands to
19 interface the decoder with the multimedia accelerator;

20 wherein the residual difference data structures and the control command
21 data structures are dynamically generated to facilitate shared media processing
22 between a decoder application executing on a host computer and a hardware
23 accelerator, communicatively coupled to the host computer based, at least in part,
24 on the auto-negotiation data structure.
25

1 **22.** An API according to claim 20, the operational data structures
2 comprising:

3 a raw bitstream data structure, dynamically generated to transfer raw media
4 content bitstream(s) to facilitate media content decoding on a hardware accelerator
5 communicatively coupled to a host computer implementing the API based, at least
6 in part, on the auto-negotiation data structure.

7
8 **23.** A storage medium comprising a plurality of executable instructions
9 which, when executed, implement an API according to claim 17.

10
11 **24.** A storage medium comprising a plurality of executable instructions
12 which, when executed, implement an application program interface (API) to
13 facilitate communication between elements of a media processing system, the API
14 including one or more auto-negotiation data structures, dynamically generated by
15 the API to negotiate at least a set of processing standards among and between one
16 or more elements of a media processing system, and one or more operational data
17 structures, dynamically generated by the API to support processing of media
18 content among and between the media processing system elements in accordance
19 with the negotiated processing standard(s).

20
21 **25.** A method facilitating media processing between elements of a
22 media processing system, the method comprising:

23 negotiating a media processing standard acceptable to each of the media
24 processing system elements from a plurality of media processing standards; and
25

1 dynamically generating operational data structures to support the negotiated
2 media processing among and between the media processing system elements.

3
4 **26.** A method according to claim 25, wherein negotiating a media
5 processing standard comprises:

6 generating auto-negotiation data structure(s) configured in accordance with
7 a proposed media processing standard;

8 issuing the auto-negotiation data structure(s) to each element of the media
9 processing system; and

10 adopting the media processing system standard if each of the elements
11 accept the proposed media processing standard.

12
13 **27.** A method according to claim 26, further comprising:
14 iteratively performing the generating and issuing steps utilizing a different
15 proposed media processing standard on subsequent iterations until an acceptable
16 media processing standard is adopted.

17
18 **28.** A method according to claim 25, wherein dynamically generating
19 operational data structures comprises:

20 generating residual difference data structure(s) to pass residual difference
21 information between media processing system elements;

22 generating control command data structure(s) to pass control commands
23 tailored in accordance with an adopted media processing standard based, at least in
24 part, on the auto-negotiation.

1 **29.** A method according to claim 25, wherein the auto-negotiation data
2 structure(s) also include a proposed split in media processing between the media
3 processing system elements.

4
5 **30.** A method according to claim 29, wherein dynamically generating
6 operational data structures comprises:

7 generating raw bitstream data structure(s), to pass raw bitstream media
8 content from a decoder application to a hardware accelerator to decode the media
9 content based, at least in part, on the negotiated split in media processing between
10 the decoder application and the hardware accelerator.

11
12 **31.** A method according to claim 29, wherein dynamically generating
13 operational data structures comprises:

14 generating a residual difference data structure, to pass residual difference
15 information between media processing system elements; and

16 generating a control command data structure, to pass control commands
17 tailored in accordance with an adopted media processing standard;

18 wherein the residual difference data structure and the control command data
19 structure are generated to facilitate shared decoding among two or more media
20 processing system elements, as negotiated between the media processing system
21 elements.

22
23 **32.** A media processing system comprising:

24 a storage medium including a plurality of executable instructions; and
25

1 an execution unit, coupled to the storage medium, to execute at least a
2 subset of the plurality of executable instructions to implement a method according
3 to claim 25.

4
5 **33.** A storage medium comprising a plurality of executable instructions
6 which, when executed, implement a method according to claim 25.

7
8 **34.** A computing system comprising:
9 a plurality of non-integrated media processing system elements, to receive
10 and process media content; and
11 an application program interface (API), communicatively coupling the non-
12 integrated media processing system elements, to automatically negotiate a media
13 processing standard acceptable to each of the plurality of non-integrated media
14 processing system standards, and to dynamically generate data structures to
15 facilitate media processing by the system elements.

16
17 **35.** A computing system according to claim 34, the non-integrated
18 media processing system elements comprising:

19 one or more media processing application(s) of a plurality of media
20 processing applications to receive and processing media content in accordance
21 with one or more media processing standards; and

22 one or more hardware accelerator(s) of a plurality of hardware accelerators
23 to perform one or more media processing tasks in accordance with one or more
24 media processing standards.
25

1 **36.** A computing system according to claim 34, wherein the API also
2 negotiates a split in media processing among the media processing system
3 elements based, at least in part, on an identified media processing capability of the
4 media processing system elements.
5

6 **37.** A computing system according to claim 36, wherein the API
7 dynamically generates a residual difference data structure to pass residual
8 difference information, and a control command data structure to pass macroblock
9 control commands tailored in accordance with an acceptable media processing
10 system standard, when two or more media processing system elements decode the
11 received media content.
12

13 **38.** A computing system according to claim 36, wherein the API
14 dynamically generates a raw bitstream data structure to pass raw bitstream media
15 content from one media processing system element to another media processing
16 system element to decode the received media content.
17

18 **39.** A computing system according to claim 38, wherein a decoder
19 application passes received media content to one or more hardware accelerators to
20 decode the media content via the raw bitstream data structure(s).
21
22
23
24
25